Proceedings of 15th Computer Vision and Pattern Recognition Conference (CVPR), June 18-20, 1996, San Francisco.

A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications

Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano and Masaya Tanaka

The Robotics Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213

abstract

We have developed a video-rate stereo machine that has the capability of generating a dense depth map at the video rate. The performance bench marks of the CMU video-rate stereo machine are: 1) multi image input of up to 6 cameras; 2) throughput of 30 million point $\times$ disparity measurement per second; 3) frame rate of 30 frame/sec; 4) a dense depth map of up to 256×240 pixels; 5) disparity search range of up to 60 pixels; 6) high precision of depth output up to 8 bits (with interpolation). The capability of passively producing such a dense depth map (3D representation) of a scene at the video rate can open up a new class of applications of 3D vision: merging real and virtual worlds in real time.

1 Introduction

Stereo range imaging uses correspondence between sets of two or more images for depth measurement. Despite a great deal of research during the past two decades, no stereo systems developed so far have achieved adequate throughput and precision to enable video-rate dense depth mapping [1,5,6,17]. The throughput of a stereo machine can be most effectively measured by the product of the number of depth measurements per second (pixels/sec) and the range of disparity search (pixels); the former determines the density and speed of depth measurement and the latter the dynamic range of distance measurement. The PRISM3 system developed by Teleos [12], the JPL stereo implemented on DataCube [10], CMU's Warp-based multi-baseline stereo [16], and INRIA's system [4] are among the most advanced real-time stereo systems; yet none of them are able to provide a complete video-rate output of range as dense as the input image with low latency.

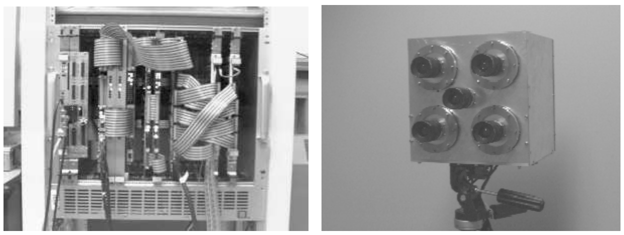
We have developed a video-rate stereo machine which has the throughput of 30 million pixel^2/sec. This throughput translates to a $200 \times 200 \times 5$bit depth image at the speed of 30 frames per second - the speed, density and depth resolution high enough to be called a video-rate 3D depth measurement camera. Our video-rate stereo machine is based on a new stereo algorithm, the multi-baseline stereo theory[13,14,11]. It uses multiple images obtained by multiple cameras to produce different baselines in lengths and in directions.

Video-rate stereo range mapping has many advantages. It is passive and it does not emit any radio or light energy. With appropriate imaging geometry, optics, and high-resolution cameras, stereo can produce a dense, precise range image. Stereo performs sensor fusion inherently; range information is aligned with visual information in the common image coordinates. Stereo depth mapping is scanless; thus it does not have the problem of apparent shape distortion from which a scanning-based range sensor suffers due to motion during a scan. These features of video-rate dense depth mapping open up a new class of applications: merging the real and virtual worlds in real time. In this paper we will present two examples, z keying and virtualized reality, on which we are currently working.

2 CMU Video-Rate Stereo Machine and Its Performance

CMU video-rate stereo machine is a special-purpose hardware that has been built with off-the-shelf components (See Figure 1). The main devices used in the machine include PLDs, high-speed ROMs, RAMs, pipeline registers, commercially available convolvers, digitizers and ALUs. All of the system is designed and built in CMU except for the video cameras, the C40 DSP array and the real-time processor board. Table 1 summarizes the current performance.

Five-eye camera head, shown in Figure 1 (b), handles the distance range of 2 to 15m using 8mm lenses. An example scene and its range image are shown in Figure 2. The stereo machine outputs a pair of intensity and depth images at 30 times/sec.



(a) Processor **(b) Five-eye camera head**

Figure 1: The CMU video-rate stereo machine

(a) intensity image **(b) corresponding disparity map**

Figure 2: An example scene and its range image

Table 1: Performance of CMU stereo machine

Number of cameras	2 to 6
Processing time/pixel	33ns × (disparity range + 2)
Frame rate	up to 30 frames/sec
Depth image size	up to 256 × 240
Disparity search range	up to 60 pixels

3 Multi-Baseline Stereo Algorithm

3.1 Theory

The stereo machine adopts the multi-baseline stereo algorithm [13]. Assuming that stereo images have been rectified, the disparity d is related to the distance z to the scene point by:

$$\frac{d}{B} = F \cdot \frac{1}{z} = \zeta \quad (1)$$

where B and F are baseline and focal length, respectively. This equation indicates that for a particular point in the image, the disparity divided by the baseline length (the inverse depth ζ) is constant since there is only one distance z for that point. If any evidence or measure of matching for the same point is represented with respect to ζ, it should consistently show a good indication only at the single correct value of ζ independent of B.

The SSD (Sum of Squared Difference) over a small window is one of the simplest and most effective measures of image matching. For a particular point in one image, a small image window is cropped around it, and it is slid along the epipolar line of other images. Suppose that the stereo camera head has a base camera f_0 and n inspection cameras $\{f_k \mid k=1,...n\}$, forming n stereo pairs. For each stereo pair we compute SSD value (SSD_k, $k=1,...n$) for a pixel (i,j) of f_0 with respect to ζ.

$$\begin{aligned} SSD_k(i,j,\zeta) &= \sum_{(s,t)\in W(i,j)} SD_k(s,t,\zeta) \\ &= \sum_{(s,t)\in W(i,j)} \left(f_k\left(s + c_1\cdot\left(B_k\cdot\zeta\right), t + c_2\cdot\left(B_k\cdot\zeta\right)\right) - f_0(s,t)\right)^2 \end{aligned} \quad (2)$$

where SD_k is the squared difference between f_0 and f_k, B_k is the baseline length between f_0 and f_k, $\mathbf{c} = (c_1,c_2)$ is the unit vector pointing the direction of the epipolar line in f_k for the pixel (i,j) of f_0 and $W(i,j)$ is a small window cropped around the position (i,j).

The curves SSD1 to SSD3 in Figure 3 show typical curves of SSD values with respect to ζ for individual stereo image pairs. Note that, as expected, these SSD functions have the same minimum position that corresponds to the true depth. We add up these SSD functions from all stereo pairs to produce the sum of SSDs, which we call SSSD-in-inverse-distance.

$$SSSD(i,j,\zeta) = \sum_{k=1}^{n} SSD_k(i,j,\zeta) = \sum_{k=1}^{n}\left(\sum_{(s,t)\in w(i,j)} SD_k(s,t,\zeta)\right) \quad (3)$$

The SSSD-in-inverse-distance has a clearer and less ambiguous minimum than individual SSDs. Also, one should notice that the valley of the SSSD curve is sharper than SSDs, meaning that we can localize the minimum position more precisely, thereby producing greater precision in depth measurement. The algorithm has been successfully tested with indoor and outdoor scenes under a variety of conditions[11,14].

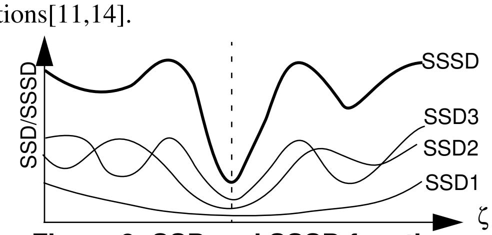


Figure 3: SSD and SSSD functions

3.2 Summary of the Algorithm

The multi-baseline stereo method consists of three steps as shown in Figure 4. The first step is the Laplacian of Gaussian (LOG) filtering of input images. This filtering enhances the image features as well as removing the effect of intensity variations among images due to difference of camera gains, ambient light, etc. The second step is the computation of SSD values for all stereo image pairs and the summation of the SSD values to produce the SSSD function. The third and final step is the identification and localization of the minimum of the SSSD function to determine the inverse depth. Uncertainty is evaluated by analyzing the curvature of the SSSD function at the minimum.

The total amount of computation per second required for the SSSD calculation is estimated as:

$$N^2 \times W^2 \times D \times (C-1) \times P \times F \qquad (4)$$

where N^2 is the image size, W^2 the window size, D the disparity range, C the number of cameras, P the number of operation per one SD calculation and F the number of frames per second. We have estimated p as 14 operations including image sampling in the subpixel precision and calculation of difference. If we set $N = 256$, $W = 11$, $D = 30$, $C = 6$, and $F = 30$, then the total computation would be 465 giga-operations. However, the most important aspect of the multi-baseline stereo algorithm is that it takes advantage of the redundancy contained in multi-stereo pairs. As a result it is a straightforward algorithm which is appropriate for hardware implementation.

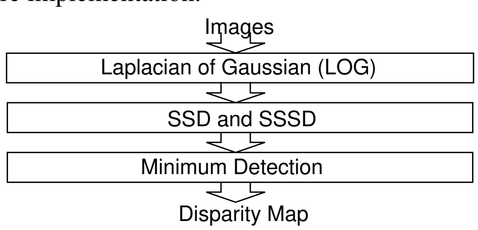


Figure 4: Outline of stereo method

4 Design of a Video-Rate Stereo Machine

The basic algorithm requires some extensions to allow for parallel, low-cost, high-speed machine implementation. The three major ones are: 1) the use of small integers for image data representation; 2) the use of absolute values instead of squares in the SSD computation (i.e., sum of absolute difference SAD instead of SSD); and 3) the capability of rectificational geometry compensation.

Figure 5 illustrates the architecture of the system developed. It consists of five subsystems: 1) multi-camera stereo head; 2) multi-image frame grabber; 3) Laplacian of Gaussian (LOG) filtering; 4) parallel computation of SSAD; and 5) subpixel localization of the minimum of the SSAD in the C40 DSP array.

These subsystems are connected to a VME Bus and controlled by a VxWorks real-time processor. System software, running on Sun workstation, enables users to utilize the machine's capabilities through a graphical interface.

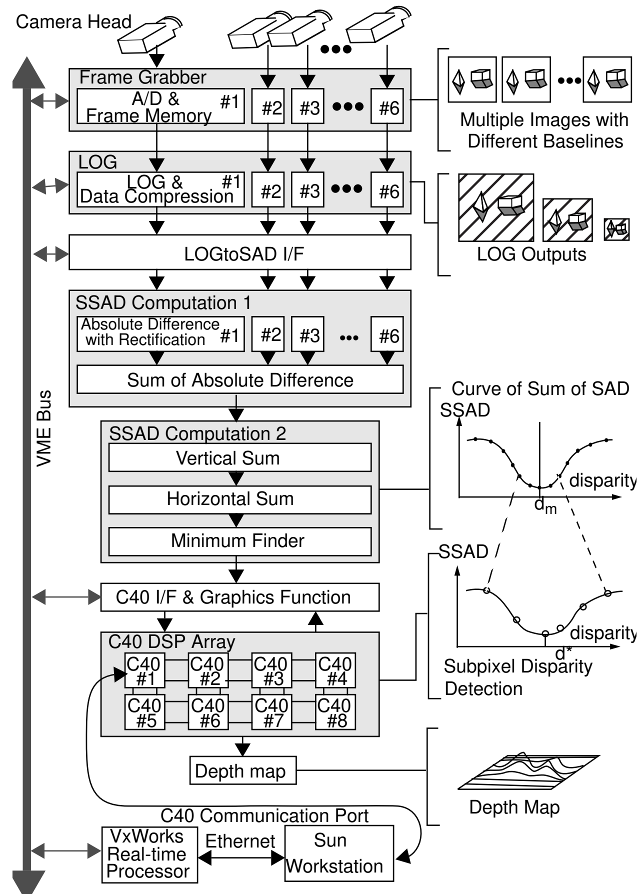


Figure 5: Architecture of stereo machine

4.1 LOG Subsystem

The LOG subsystem contains six channels of processor, each of which can perform the Laplacian of Gaussian (LOG) filtering on an image at video rate. The input image for each channel is read from the frame grabber and the output image is sent to the SSAD subsystems. Figure 6 shows the function of the LOG subsystem for one channel. Four 7×7 convolvers are used for each channel. By loading arbitrary 7×7 coefficient we can realize a large class of filtering operations. For example, a LOG filtering is achieved by loading a Gaussian mask into the first three convolvers and a Laplacian filter into the final one. The maximum size of LOG filter becomes 25×25 by this cascading technique. The LOG subsystem also has a multi-resolution capability which produces an image pyramid by repeatedly shrinking the images [2].

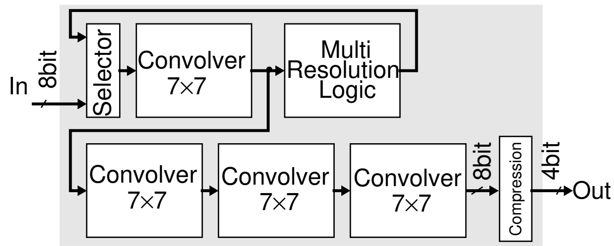


Figure 6: Function of LOG subsystem

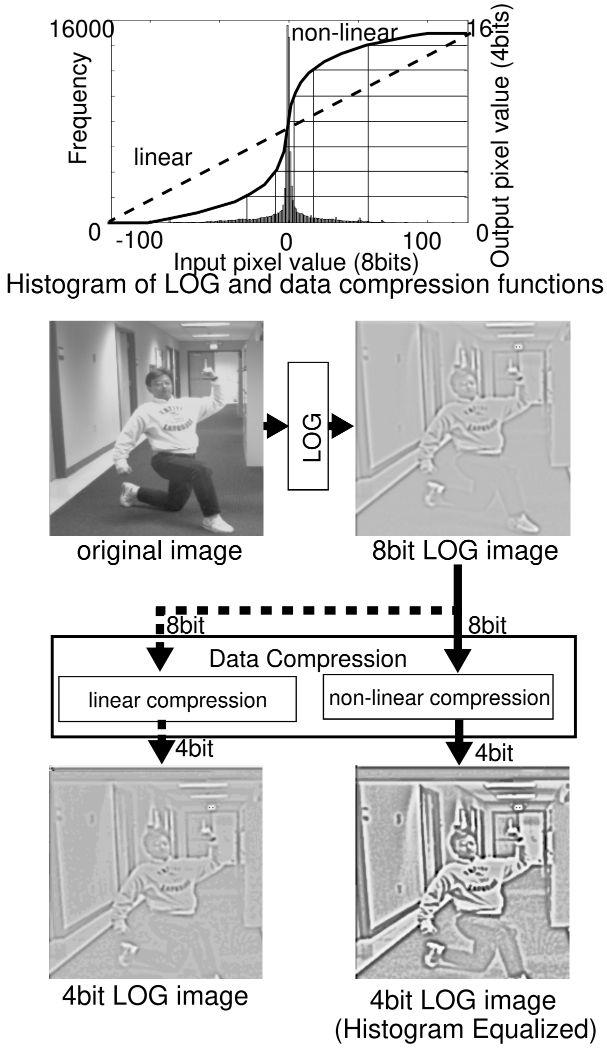


Figure 7: 8bit to 4bit Data Compression of LOG image

After the LOG filtering, we compress the output data from 8 bits to 4 bits, primarily to reduce the hardware size of the SSAD subsystem which follows this stage. A typical example of the histogram of output values of LOG filtering in 8 bits is shown at the top of Figure 7. The distribution of the pixel values typically concentrates around zero. With such a distribution, linear data compression would put most of pixels into the same value and most features would be lost. Instead, we use nonlinear compression which approximates the effect of histogram equalization. The two images of 4bit LOG at the middle of Figure 7 show the difference between the two types of compression of LOG data. The output of the nonlinear compression has more features because it enables the data values closer to zero to be represented more finely, while values further from zero are divided more coarsely.

In software experiments, we confirmed that there was not much difference between the disparity map calculated with 8 bit data and the disparity map calculated with 4 bit data which are obtained using a histogram equalization technique. In the final stereo machine hardware, we use a built-in table for conversion instead of computing a histogram for each image on the fly.

4.2 SSAD Subsystem

The SSAD subsystem has three functions. First, it rectifies the images which come from LOG output. The SSAD calculation follows the rectification. The method of SSAD calculation is optimized for a compact hardware implementation. Finally, the minimum finder detects the disparity value which minimizes the SSAD value. These functions are implemented on two VME bus boards.

4.2.1 Rectification of Images

The calculation of squared difference SD_k in equation (2) assumes inputs of rectified images. In general, however, since multiple stereo cameras are not perfectly aligned, and/or optical systems are not perfect, video rate image rectification and correction are required.

Suppose we have multiple images $\{f_k \mid k=0,...n\}$ which are not rectified. Then the absolute difference $AD_k(s, t, \zeta)$, instead of the squared difference $SD_k(s, t, \zeta)$, has the following expression.

$$AD_k(s, t, \zeta) = f_k\left(I_k(s, t, \zeta), J_k(s, t, \zeta)\right) - f_0\left(I_0(s, t), J_0(s, t)\right) \quad (5)$$

Here I_k and J_k are functions of rectified coordinates (s,t) and ζ, while I_0 and J_0 are functions of only (s,t). Either strong calibration methods [15, 9] or weak calibration methods [3] enable us to obtain these functions.

The SSAD subsystem stores these functions in RAM in the form of tables. Using these tables, the SSAD hardware calculates absolute differences in the rectified coordinates (see Figure 8). The tables are obtained at the time of calibration and are loaded when the machine starts up.

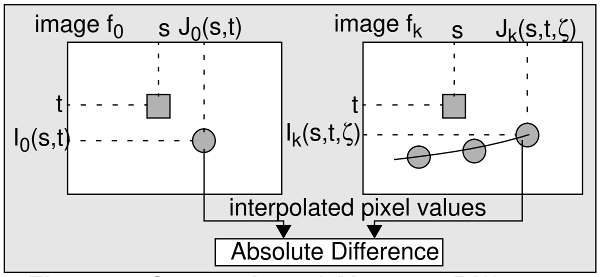


Figure 8: Calculation of Absolute Difference with Rectification

4.2.2 Optimized SSAD Calculation

The number of SSAD (or SSSD) operations evaluated by the formula (4) includes redundancy of absolute difference (or squared difference) calculation which is repeated W^2 times with the same combination of coordinates (s,t) and inverse depth ζ. This redundancy can be removed by changing the order of summation in equation (3) and expressing the window summation in a recursive manner.

By converting square difference (SD) to absolute difference (AD), and changing the order of summation, we can rewrite equation (3) as:

$$SSAD(i,j,\zeta) = \sum_{(s,t) \in W(i,j)} \left(\sum_{k=1}^{n} \left(AD_k(s,t,\zeta) \right) \right) \quad (6)$$

Assuming that $i - m \le s \le i + m$ and $j - m \le t \le j + m$ $(m \ge 0)$, we get the following expression for the SSAD.

$$SSAD(i,j,\zeta) = \sum_{s=i-m}^{i+m} \left(\sum_{t=j-m}^{j+m} \left(\sum_{k=1}^{n} \left(AD_k(s,t,\zeta) \right) \right) \right) \quad (7)$$

The first and the second summations correspond to the horizontal and vertical summations within a window, respectively. Let the second sum be denoted by $VSUM(i,j,\zeta)$:

$$VSUM(i,j,\zeta) = \sum_{t=j-m}^{j+m} \left(\sum_{k=1}^{n} \left(AD_k(i,t,\zeta) \right) \right) \quad (8)$$

Equation (8) can be written in a recursive form.

$$SSAD(i,j,\zeta) = SSAD(i-1,j,\zeta) - VSUM(i-m-1,j,\zeta) + VSUM(i+m,j,\zeta) \quad (9)$$

Similarly, $VSUM$ itself can also be written in a recursive form.

$$VSUM(i,j,\zeta) = VSUM(i,j-1,\zeta) - \sum_{k=1}^{n} \left(AD_k(i,j-m-1,\zeta) \right) + \sum_{k=1}^{n} \left(AD_k(i,j+m,\zeta) \right) \quad (10)$$

The SSAD calculation with the equations (9) and (10) eliminates redundancy of calculating absolute difference with the same combination of coordinates and inverse depth. Therefore, we could save memory space and other hardware components to result in a compact hardware implementation.

4.2.3 Minimum Finder

The minimum finder module, located at the end of SSAD subsystem, selects the minimum value and its position in the SSAD function together with its neighboring SSAD values, and transfers them to the C40 DSP array.

4.3 Subpixel Disparity Detection

The C40 DSP array performs sub-pixel interpolation of disparity and uncertainty estimation using quadratic function fitting around the minimum value. This extends the disparity resolution to 8 bits. Figure 9 demonstrates a result of subpixel interpolation of disparity. For the scene (a), the image (b) shows its depth map with a disparity range of 30 (approximately 5 bits). The interpolated depth map (8 bits) shown in the image (c) has smoother graduation than (b). Currently disparity measurement with interpolation operates at 15 frames per second with the frame size of 200×200 image size.

(a) **(b)** **(c)**

Figure 9: Example scenes demonstrating the performance of subpixel interpolation of depth

(a) an intensity image
(b) the corresponding depth map with 30 disparity range
(c) the interpolated depth in a precision of 8 bits

4.4 A Camera Head

A stereo head with 5 CCD cameras has been built (see Figure 1(b)). The camera at the middle of the camera head is the base camera f_0, with which the other cameras make four stereo pairs. The symmetrical arrangement of cameras helps to reduce effects of occlusion because each pixel of the image of the base camera can be seen in at least one of the other four camera images. Figure 10 illustrates the effect of using multiple cameras for stereo. Figure 10 (b) shows depth map of the machine when using only two stereo pairs on the right hand side of the base camera. Occlusions result in noisier depth measurement at the right side of human body. Using all four symmetric stereo pairs (Figure 10 (c)) improves the result substantially.

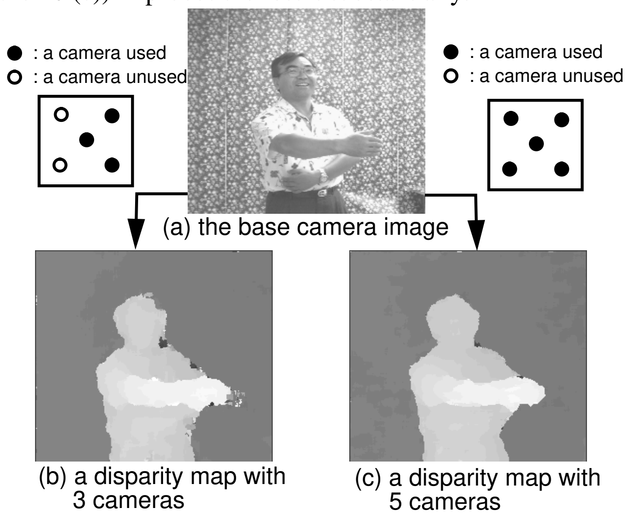


Figure 10: Example scenes of disparity map with occlusions and without occlusions

5 New Applications of the Stereo Machine

Besides robotic applications, such as autonomous vehicles, there are many other applications of the stereo machine. The capability of producing a dense 3D representation at video rate opens up a new class of applications for 3D vision. We have been working on two such new applications: virtualized reality [7] and z keying.

5.1 Z Keying

In visual media communication and display, it is often necessary to merge a video signal from a real camera and a synthetic video signal from computer graphics. Chroma keying is a standard technique for such a purpose, as used in TV weather reports. A weather man is imaged by a real camera in front of a blue screen, and the pixels which have blue color, that is, the portions of the scene that are not occluded by the real objects, are replaced by the synthetic image. Chroma keying, therefore, implicitly assumes that a real world object is in front of the synthetic world. Z keying is a new technique for merging real and virtual world images in a more flexible way. It uses the depth information, instead of chromaticity, as the key for switching between images. Figure 11 illustrates the idea with a real example. The depth value from the real world (the output of the stereo machine) is compared pixel by pixel with that of the virtual world (the z buffer from the graphic system), and the pixel color (or intensity) of the world closer to the camera is selected for display. As a result, real world objects can be placed in any desired relationship with virtual world objects. As shown in the example of Figure 11, part of the real object (e.g., hand) occludes the virtual objects (e.g., lamp), which in turn occludes the real objects (e.g., body). Currently, our system can perform z keying in real time at 15 frames per second.

5.2 Virtualized Reality

Once a depth map is obtained (or actually, once pixel-wise correspondences are established between images), we can place a virtual (soft) camera at places other than the original camera position, and compute the image that it would generate (except the portions that are occluded in the original views). To reduce the occluded area of the scene, we can think of a dome which is fully covered by a number of cameras. A real-time-varying event is captured or tran-

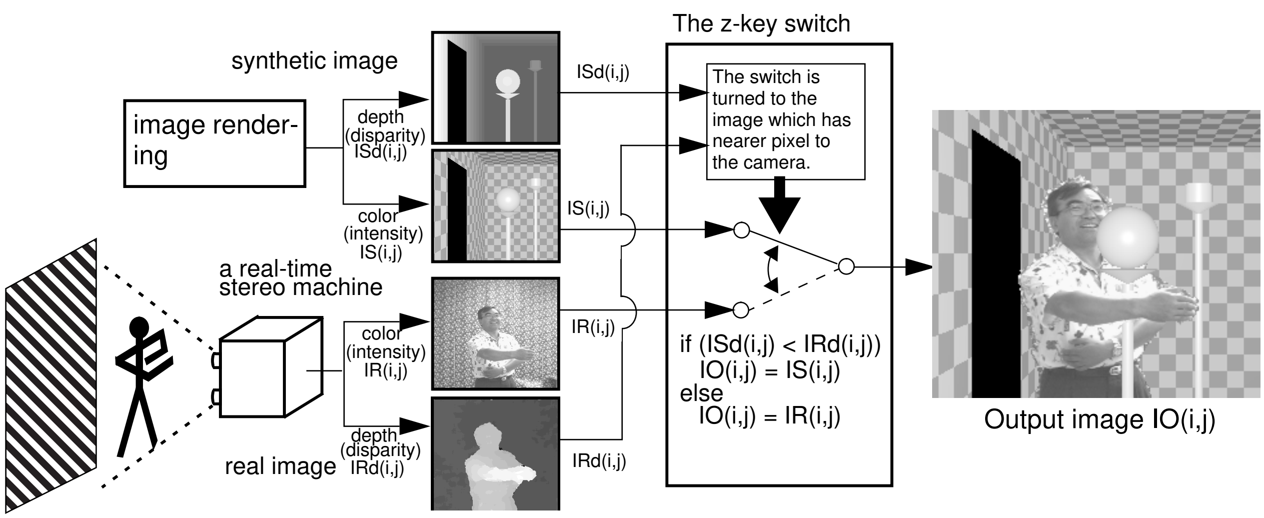


Figure 11: The scheme for z keying

scribed by those cameras, and then its 3D structure is recovered. Once the event is "virtualized" this way, a user, wearing a stereo viewer, can freely move about in the space and observe the event from any position or angle. We have built a prototype system of such a 3D Virtualization Studio. It consists of a hemispherical dome, 5 meters in diameter, and is currently populated with 51 cameras. Figure 12 shows an example of a synthesized image sequence of a virtualized "baseball" scene. A scene of a person swinging a bat is captured, and the ball's eye view is hit by the bat, and soars high and away into the sky[8]. Due to the limitations in image input and computation, this example was created off-line.

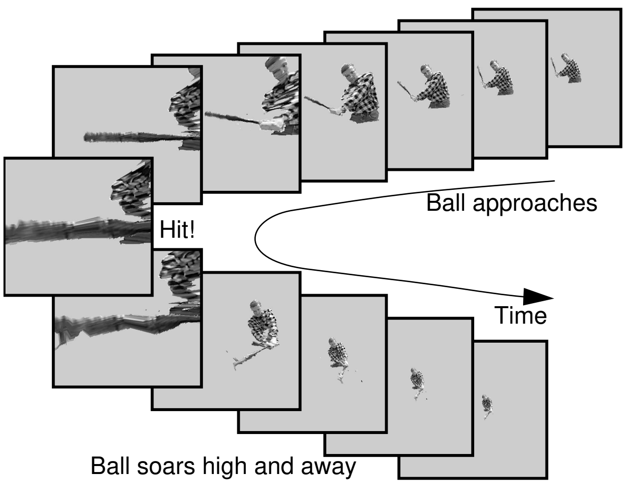


Figure 12: A "baseball" sequence from the ball's point of view

6 Conclusion

This paper has presented the CMU video-rate stereo machine and a couple of its applications. The machine is capable of producing a dense 200×200 depth map, aligned with intensity information, at 30 frames per second. This performance represents a one or two order of magnitude improvement over the current state of the art in passive stereo range mapping. Such a capability opens up a new class of applications of 3D vision, and we have briefly presented two examples in the area of visual media interaction.

Acknowledgments

We express thanks to P J Narayanan and Peter Rander for offering Figure 12. We express thanks to Larry Lyle for his help in the development of frame grabber board.

References

[1] Nicholas Ayache and Francis Lustman, Trinocular stereovision for robotics. Technical Report 1086, INRIA, Sept. 1989.

[2] P.J.Burt and E.H.Adelson, The Laplacian Pyramid as a Compact Image Code, IEEE Trans. on Communication, Vol.COM-31, No.4,pp.532-540.

[3] Olivier Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig?, In Computer Vision - ECCV '92, LNCS-Series Vol. 588, Springer - Verlag, pages 563-578, 1992.

[4] Olivier Faugeras, et al., Real time correlation based stereo: algorithm, implementations and applications, Research Report 2013, INRIA Sophia-Antipolis, 1993.

[5] Pascal Fua, A parallel stereo algorithm that produces dense depth maps and preserves image features. Technical Report 1369, Unite de Recherche, INRIA-Sophia Antipolis, France, January 1991.

[6] Ali E.Kayaalp and James L. Eckman, A pipeline architecture for near real-time stereo range detection. Technical Report GDLS-AI-TR-88-1, General Dynamics AI Lab, November 1988.

[7] Takeo Kanade, P.J. Narayanan and Peter Rander, Virtualized Reality: Concepts and Early Results. In Proc. of IEEE workshop on the Representation on Visual Scene, Boston, June 25, 1995.

[8] Takeo Kanade, P.J. Narayanan and Peter Rander, Virtualized (Not Virtual) Reality. (to be presented) In Proc. of A presentation is schedule at the 15th International Display Research Conference (Asia Display 95), Oct 16-18, 1995, Hamamatsu, Japan.

[9] S.Kimura, T.Kanade, H.Kano, A.Yoshida, E.Kawamura and K.Oda, CMU Video-Rate Stereo Machine, Mobile Mapping Symposium, May 24-26, 1995, Columbus, OH.

[10] L.H.Matthies, Stereo vision for planetary rovers: stochastic modeling to near real time implementation. International Journal of Computer Vision, 8 (1):71-91,1992.

[11] T.Nakahara and T.Kanade, Experiments in multiple-baseline stereo. Technical report, Carnegie Mellon University, Computer Science Department, August 1992.

[12] H.K.Nishihara, Real-time implementation of a sign-correlation algorithm for image-matching. (Draft) Teleos Research, February 1990.

[13] Masatoshi Okutomi and Takeo Kanade, A multi-baseline stereo. In Proc. of Computer Vision and Pattern Recognition, June 1991. Also appeared in IEEE Trans. on PAMI, 15(4),1993.

[14] Masatoshi Okutomi, Takeo Kanade and N.Nakahara, A multiple-baseline stereo method. In Proc. of DARPA Image Understanding Workshop, pages 409-426. DARPA, January 1992.

[15] Roger Y.Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE Journal of Robotics and Automation, Vol.RA-3, No.4, August 1987.

[16] J.Webb, Implementation and performance of fast parallel multi-baseline stereo vision. In Proc. of Image Understanding Workshop, pages 1005-1012. DARPA, April 1993.

[17] Kazuhiro Yoshida and Hirose Shigeo, Real-time stereo vision with multiple arrayed camera. Tokyo Institute of Technology, Department of Mechanical Engineering Science, 199X.